



NAME: VOMMIDAPU CHINNI

BATCH NO: 120

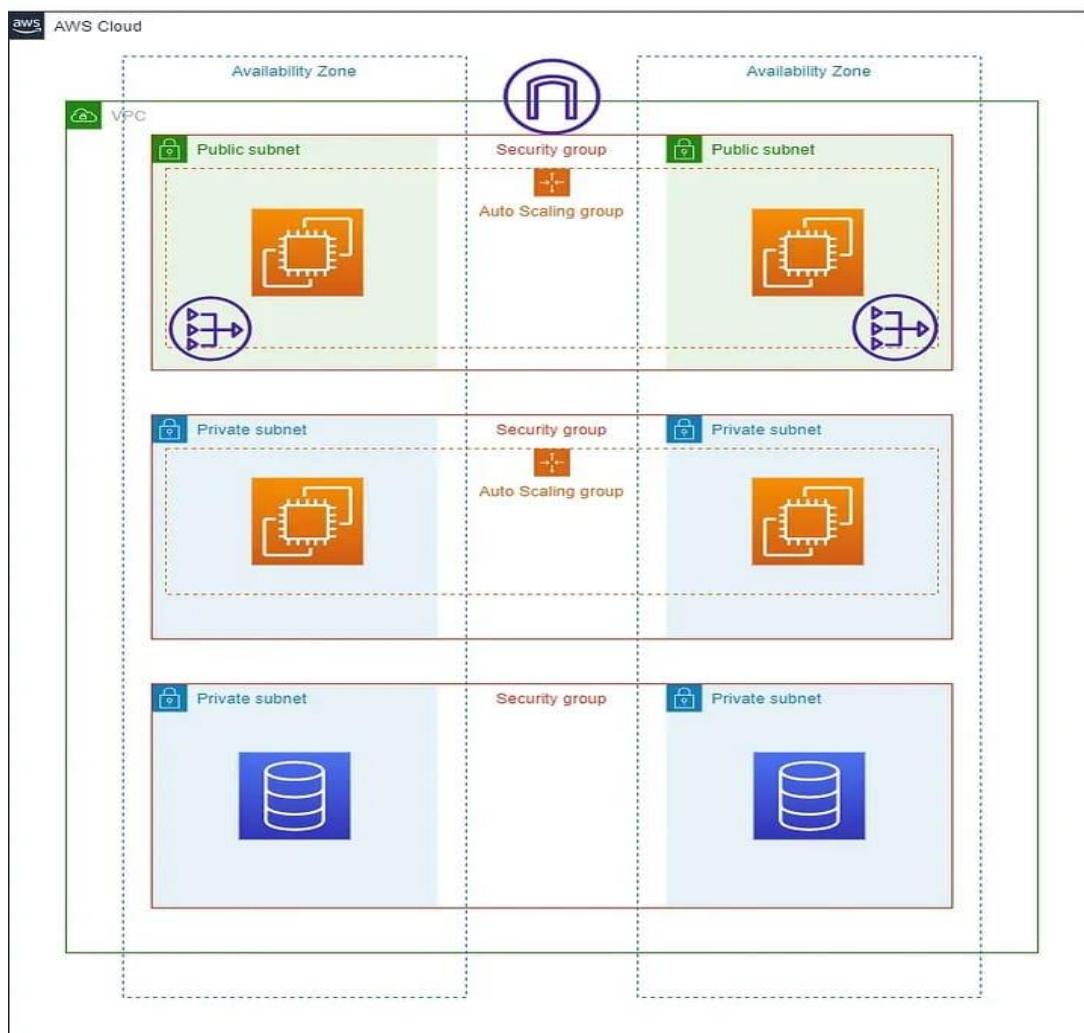
MAIL.ID: vommidapuchinni@gmail.com

DATE: 19/03/24

TRAINER: Mr. Madhukar

PROJECT – 1

Project:



The above architecture is the “Architecting 3 Tier Architecture on AWS”.

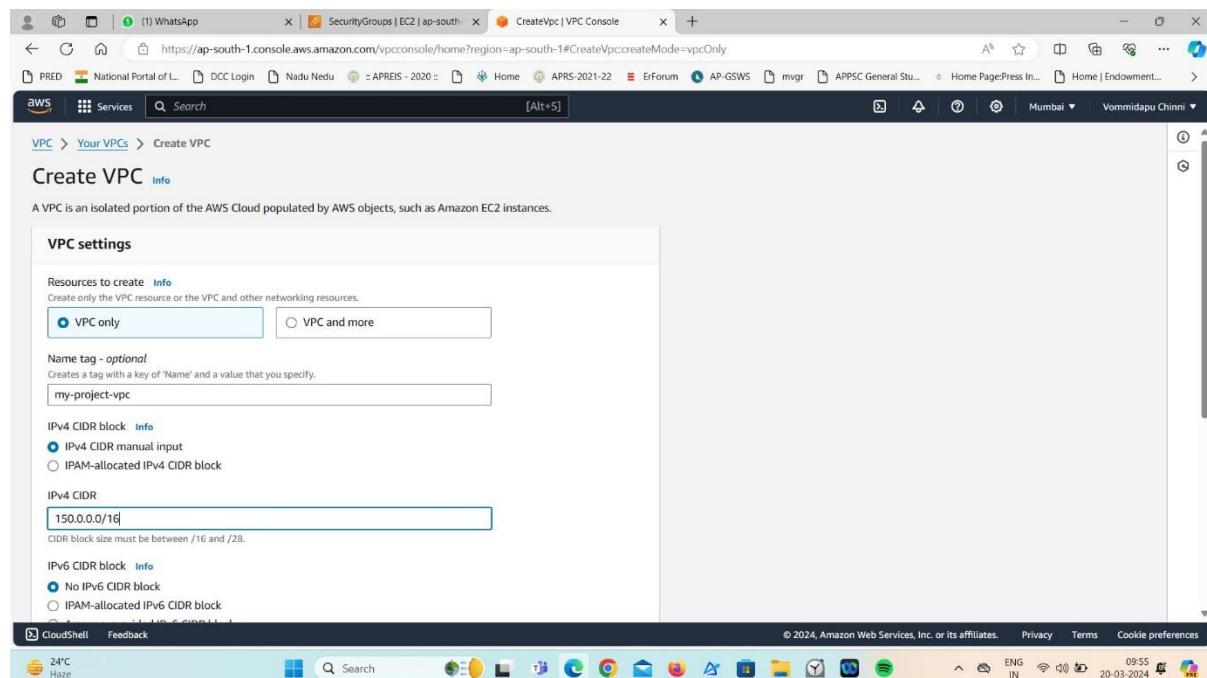
- First tier of our architecture is a web tier. It consists of 2 public subnets in separate availability zones, and an Auto scaling group with launch template and security group.
- The second tier is an Application tier. This tier will consist of 2 Private subnets, an Autoscaling group with launch a template and same security group used in above web tier.
- The third tier is a Database tier. This tier will have an RDS(relational database service) in 2 Private subnets and an same security group used in above both the tiers.

Creating the above architecture we have to follow the following steps:

1. Create VPC, Subnets – 6, Internet gate way – 1, Route tables – 2, Nat gate way – 1.
2. Launch an EC2 instance.
3. Create an AMI (image).
4. Create Autoscaling group, Create launch template.
5. Create Subnet group.
6. Create Database (RDS).
7. Establish connection.

Step 1:

Create VPC and its components:

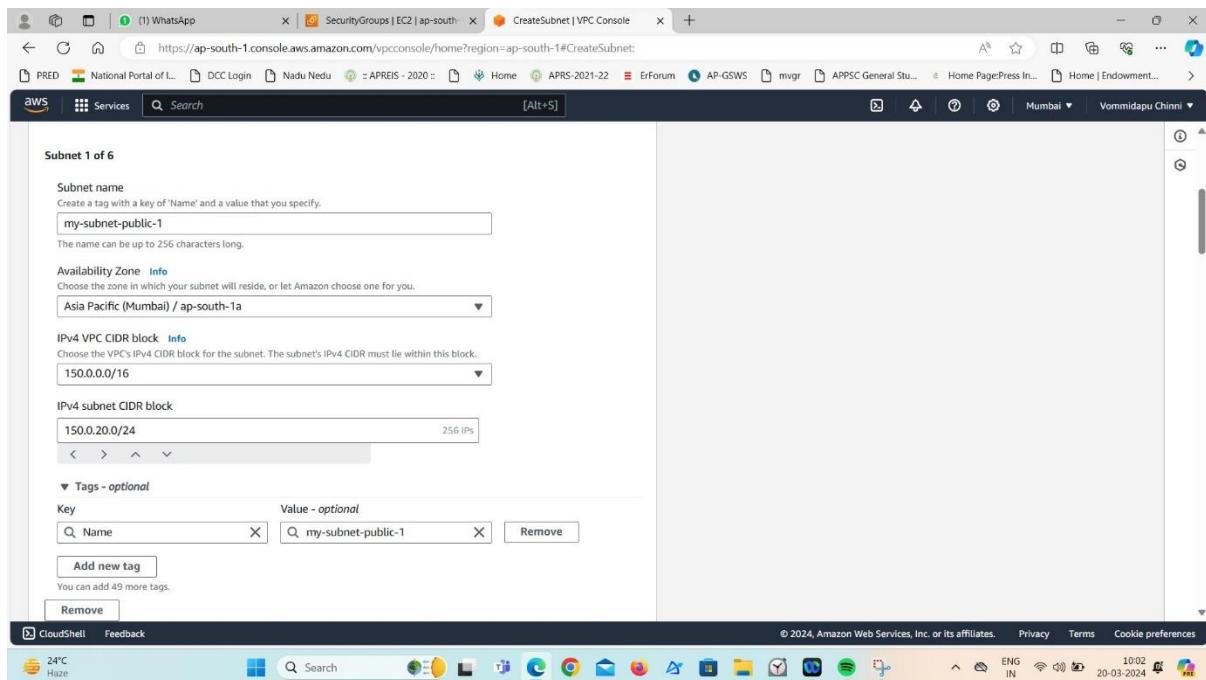


- Go to VPC dashboard click on create VPC.
- Click on VPC only and name tag as my-project-vpc.
- Give IPV4 CIDR (classless inter domain routing) as 150.0.0.0/16.
- Click on VPC, it is created.

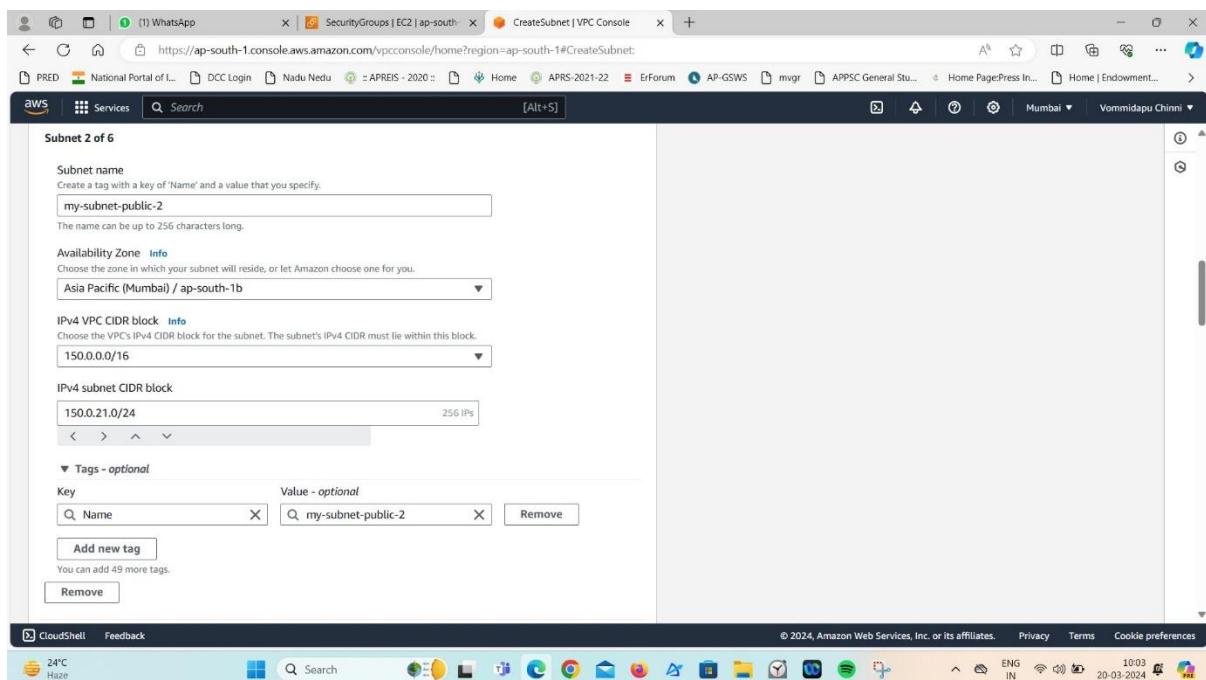
The screenshot shows the AWS VPC console interface. On the left, there's a navigation sidebar with options like VPC dashboard, EC2 Global View, Filter by VPC, Virtual private cloud, Security, and CloudShell. The main area displays a table titled "Your VPCs (1/2) info". The table has columns for Name, VPC ID, State, IPv4 CIDR, IPv6 CIDR, and DHC. There are two entries: one for a VPC with a long ID and another for "my-project-vpc" which has a shorter ID and a CIDR of 150.0.0.0/16. Below the table, a specific VPC entry is expanded, showing its details: VPC ID (vpc-0d7f16b48e6ae8853), State (Available), DHCP option set (dopt-0673a7fc1667054c), DNS hostnames (Enabled), and Main route table (rtb-00ecc7b1dfa7456c2). The bottom of the screen shows the Windows taskbar with various icons and the date/time (20-03-2024).

- Create 6 subnets (2-public, 4-private).
- Create first subnet.
- Click on subnet, click on create subnet, select our VPC (my-project-vpc).
- Give name tag as my-subnet-public-1, select availability zone as ap-south-1a. Give CIDR as 150.0.20.0/24 and created it.

The screenshot shows the "Create subnet" wizard. Step 1: "VPC" - VPC ID is selected as "vpc-0d7f16b48e6ae8853 (my-project-vpc)". Step 2: "Associated VPC CIDRs" - IPv4 CIDR is listed as "150.0.0.0/16". Step 3: "Subnet settings" - Subnet name is "my-subnet-public-1". The bottom of the screen shows the Windows taskbar with various icons and the date/time (20-03-2024).



- Create second subnet. Subnet name as my-subnet-public-2
- Availability zone as ap-south-1b.
- Give IPV4 CIDR as 150.0.21.0/24.
- Create the second subnet.



- Create third subnet. Subnet name as my-subnet-private-1.
- Availability zone as ap-south-1a.
- Give IPV4 CIDR as 150.0.22.0/24.

- Create the third subnet.

The screenshot shows the 'CreateSubnet | VPC Console' page. The 'Subnet 3 of 6' section is active. The 'Subnet name' field contains 'my-subnet-private-1'. The 'Availability Zone' dropdown is set to 'Asia Pacific (Mumbai) / ap-south-1a'. The 'IPv4 VPC CIDR block' dropdown is set to '150.0.0.0/16'. The 'IPv4 subnet CIDR block' dropdown is set to '150.0.22.0/24'. Under 'Tags - optional', there is a single tag named 'Name' with value 'my-subnet-private-1'. The status bar at the bottom shows 'CloudShell Feedback' and system information like '24°C Haze' and 'ENG IN 10:03 20-03-2024'.

- Create fourth subnet. Subnet name as my-subnet-private-2
- Availability zone as ap-south-1b.
- Give IPV4 CIDR as 150.0.23.0/24.
- Create the fourth subnet.

The screenshot shows the 'CreateSubnet | VPC Console' page. The 'Subnet 4 of 6' section is active. The 'Subnet name' field contains 'my-subnet-private-2'. The 'Availability Zone' dropdown is set to 'Asia Pacific (Mumbai) / ap-south-1b'. The 'IPv4 VPC CIDR block' dropdown is set to '150.0.0.0/16'. The 'IPv4 subnet CIDR block' dropdown is set to '150.0.23.0/24'. Under 'Tags - optional', there is a single tag named 'Name' with value 'my-subnet-private-2'. The status bar at the bottom shows 'CloudShell Feedback' and system information like '24°C Haze' and 'ENG IN 10:04 20-03-2024'.

- Create fifth subnet. Subnet name as my-subnet-private-3

- Availability zone as ap-south-1c.
- Give IPV4 CIDR as 150.0.24.0/24.
- Create the fifth subnet.

Subnet 5 of 6

Subnet name
Create a tag with a key of 'Name' and a value that you specify.

The name can be up to 256 characters long.

Availability Zone [Info](#)
Choose the zone in which your subnet will reside, or let Amazon choose one for you.

IPv4 VPC CIDR block [Info](#)
Choose the VPC's IPv4 CIDR block for the subnet. The subnet's IPv4 CIDR must lie within this block.

IPv4 subnet CIDR block
 256 IPs
< > ^ v

Tags - optional
Key Value - optional

You can add 49 more tags.

- Create sixth subnet. Subnet name as my-subnet-private-4
- Availability zone as ap-south-1a.
- Give IPV4 CIDR as 150.0.25.0/24.
- Create the sixth subnet.

Subnet 6 of 6

Subnet name
Create a tag with a key of 'Name' and a value that you specify.

The name can be up to 256 characters long.

Availability Zone [Info](#)
Choose the zone in which your subnet will reside, or let Amazon choose one for you.

IPv4 VPC CIDR block [Info](#)
Choose the VPC's IPv4 CIDR block for the subnet. The subnet's IPv4 CIDR must lie within this block.

IPv4 subnet CIDR block
 256 IPs
< > ^ v

Tags - optional
Key Value - optional

You can add 49 more tags.

The screenshot shows the AWS VPC Subnets page. On the left, there's a navigation sidebar with sections like EC2 Global View, Filter by VPC, Virtual private cloud, Subnets, Route tables, Internet gateways, Egress-only internet gateways, DHCP option sets, Elastic IPs, Managed prefix lists, Endpoints, Endpoint services, NAT gateways, and Peering connections. The main area displays a table titled 'Subnets (9) Info' with columns: Name, Subnet ID, State, VPC, and IPv4 CIDR. The subnets listed are:

Name	Subnet ID	State	VPC	IPv4 CIDR
my-subnet-private-1	subnet-0e1bf9607ad2d1b57	Available	vpc-0d7f16b48e6ae8853 my...	150.0.22.0/24
my-subnet-public-1	subnet-043c9ccff6f4a7326	Available	vpc-0d7f16b48e6ae8853 my...	150.0.20.0/24
my-subnet-public-2	subnet-009125c4cf54052ee	Available	vpc-0d7f16b48e6ae8853 my...	150.0.21.0/24
my-subnet-private-2	subnet-06118ba884b21366b	Available	vpc-0d7f16b48e6ae8853 my...	150.0.23.0/24
my-subnet-private-4	subnet-0b406c4c29dc4489f	Available	vpc-0d7f16b48e6ae8853 my...	150.0.25.0/24
my-subnet-private-3	subnet-023f7d550de292ae6	Available	vpc-0d7f16b48e6ae8853 my...	150.0.24.0/24

At the bottom, there's a section titled 'Select a subnet' with a dropdown menu.

These are the subnets we created.

For both public subnets and enable auto assign IP setting in edit subnet settings.

The screenshot shows the 'Edit subnet settings' page for subnet-043c9ccff6f4a7326. The page has several sections:

- Subnet**: Shows Subnet ID (subnet-043c9ccff6f4a7326) and Name (my-subnet-public-1).
- Auto-assign IP settings**:
 - Enable auto-assign public IPv4 address
 - Enable auto-assign customer-owned IPv4 address (Info: Option disabled because no customer owned pools found.)
- Resource-based name (RBN) settings**:
 - Enable resource name DNS A record on launch
 - Enable resource name DNS AAAA record on launch
- Hostname type**:

At the bottom, there's a footer with standard AWS links and a status bar showing 24°C Haze, ENG IN, 10:06, 20-03-2024, and a colorful icon.

The screenshot shows the 'Edit subnet settings' page for a subnet named 'my-subnet-public-2'. Key details include:

- Subnet ID:** subnet-009125c4cf54052ee
- Name:** my-subnet-public-2
- Auto-assign IP settings:**
 - Enable auto-assign public IPv4 address:** Checked.
 - Enable auto-assign customer-owned IPv4 address:** Option disabled because no customer owned pools found.
- Resource-based name (RBN) settings:**
 - Enable resource name DNS A record on launch:** Unchecked.
 - Enable resource name DNS AAAA record on launch:** Unchecked.
 - Hostname type:** Info

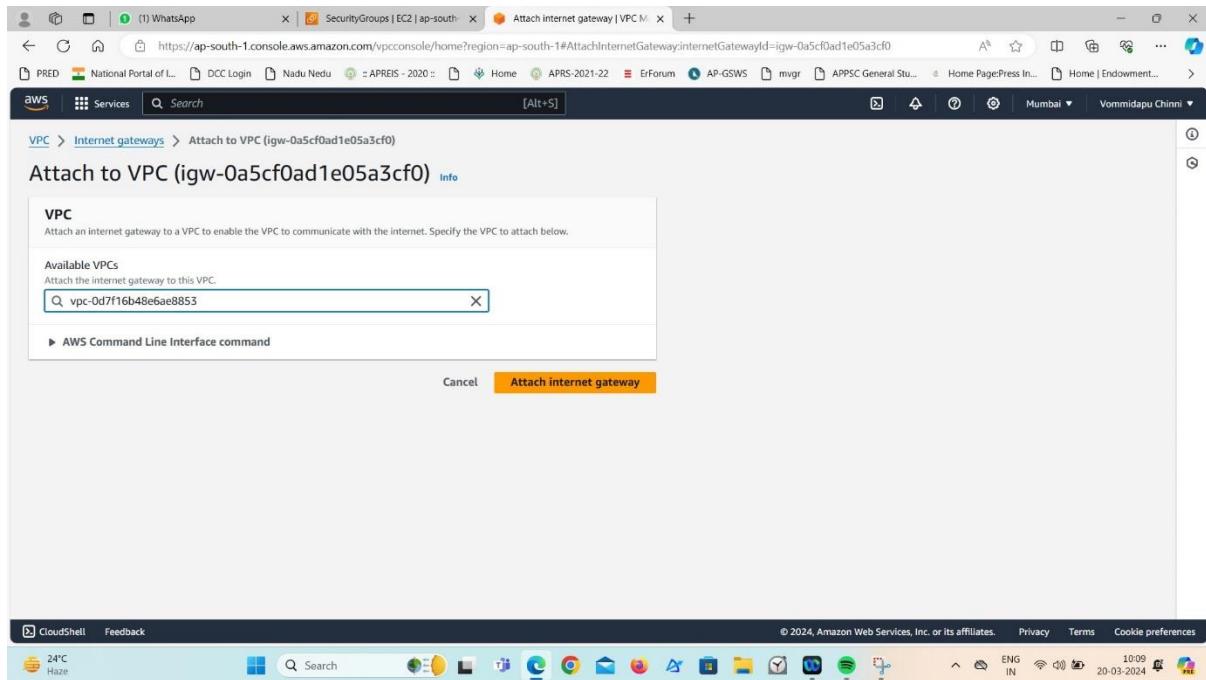
- Create internet gateway, name tag as my-igw

The screenshot shows the 'Create internet gateway' page. Configuration includes:

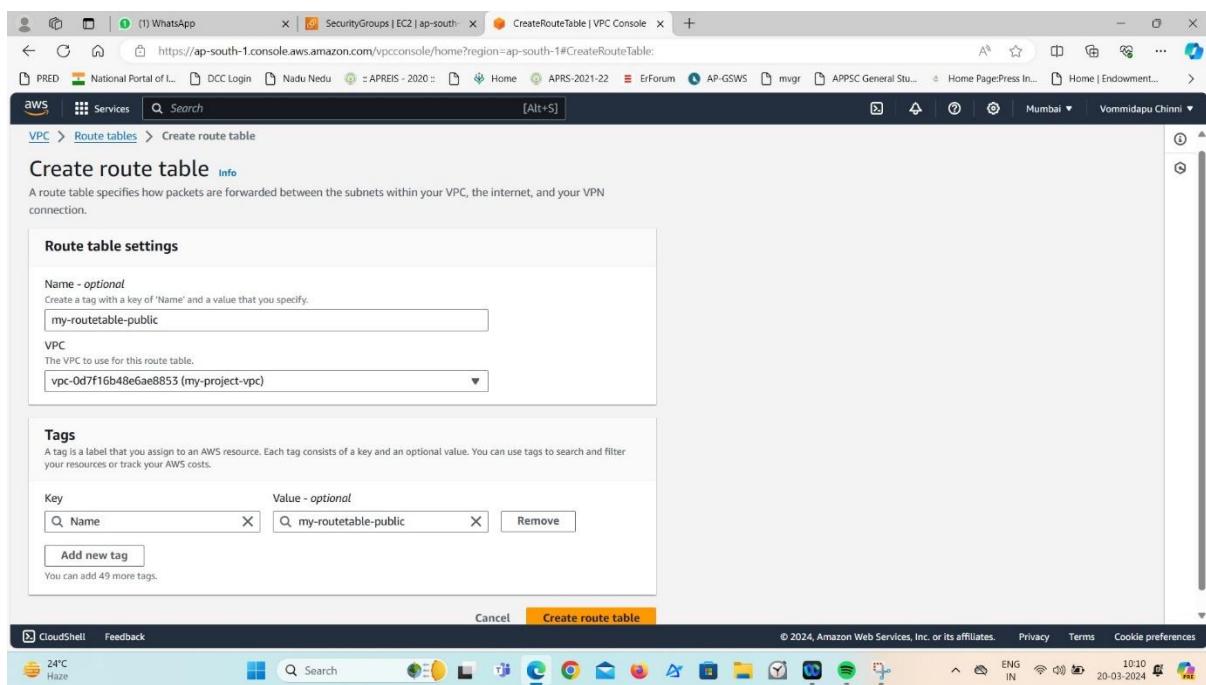
- Name tag:** my-igw
- Tags - optional:**
 - Key:** Name, **Value - optional:** my-igw
 - Add new tag:** You can add 49 more tags.

At the bottom, there are 'Cancel' and 'Create internet gateway' buttons.

- This igw is attached to VPC.
- Go to actions in internet gate way and click on attach to VPC.
- Select our VPC (my-project-vpc). Click on attach internet gateway.



- Create route table, give name as my-routetable-public
- Select our VPC (my-project-vpc), create it.



- Click on route table id, open it.
- Go down click on edit subnet association.
- Select both public subnet and click on save association.

Available subnets (2/6)

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	Route table ID
my-subnet-private-3	subnet-0fdcb345c9cf34e70	150.0.24.0/24	-	Main (rtb-00ecc7b1dfa7456c2)
my-subnet-private-1	subnet-0e1bf9607ad2d1b57	150.0.22.0/24	-	Main (rtb-00ecc7b1dfa7456c2)
<input checked="" type="checkbox"/> my-subnet-public-1	subnet-043c9ccff6f4a7326	150.0.20.0/24	-	Main (rtb-00ecc7b1dfa7456c2)
<input checked="" type="checkbox"/> my-subnet-public-2	subnet-009125c4cf54052ee	150.0.21.0/24	-	Main (rtb-00ecc7b1dfa7456c2)
my-subnet-private-2	subnet-06118ba884b21366b	150.0.23.0/24	-	Main (rtb-00ecc7b1dfa7456c2)
my-subnet-private-4	subnet-0b406c4c29dc4489f	150.0.25.0/24	-	Main (rtb-00ecc7b1dfa7456c2)

Selected subnets

- [subnet-043c9ccff6f4a7326 / my-subnet-public-1](#)
- [subnet-009125c4cf54052ee / my-subnet-public-2](#)

Actions: Cancel **Save associations**

- Go to actions click on edit routes.
- Click on add routes give all traffic (0.0.0.0/0) and select our internet gateway, save changes.

Edit routes

Destination	Target	Status	Propagated
150.0.0.0/16	local	Active	No
0.0.0.0/0	Internet Gateway	-	No

Actions: Add route Cancel Preview Save changes

- Create private route table name as my-routetable-private.
- Select our VPC, (my-project-vpc), create it.
- Click on route table id, open it.
- Go down click on edit subnet association.
- Select all private subnet and click on save association.

Create route table info

A route table specifies how packets are forwarded between the subnets within your VPC, the internet, and your VPN connection.

Route table settings

Name - *optional*
Create a tag with a key of 'Name' and a value that you specify.

my-routetable-private

VPC
The VPC to use for this route table.

vpc-0d7f16b48e6ae8853 (my-project-vpc)

Tags

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key	Value - <i>optional</i>
Q Name	Q my-routetable-private

Add new tag
You can add 49 more tags.

Cancel **Create route table**

Edit subnet associations

Change which subnets are associated with this route table.

Available subnets (4/6)

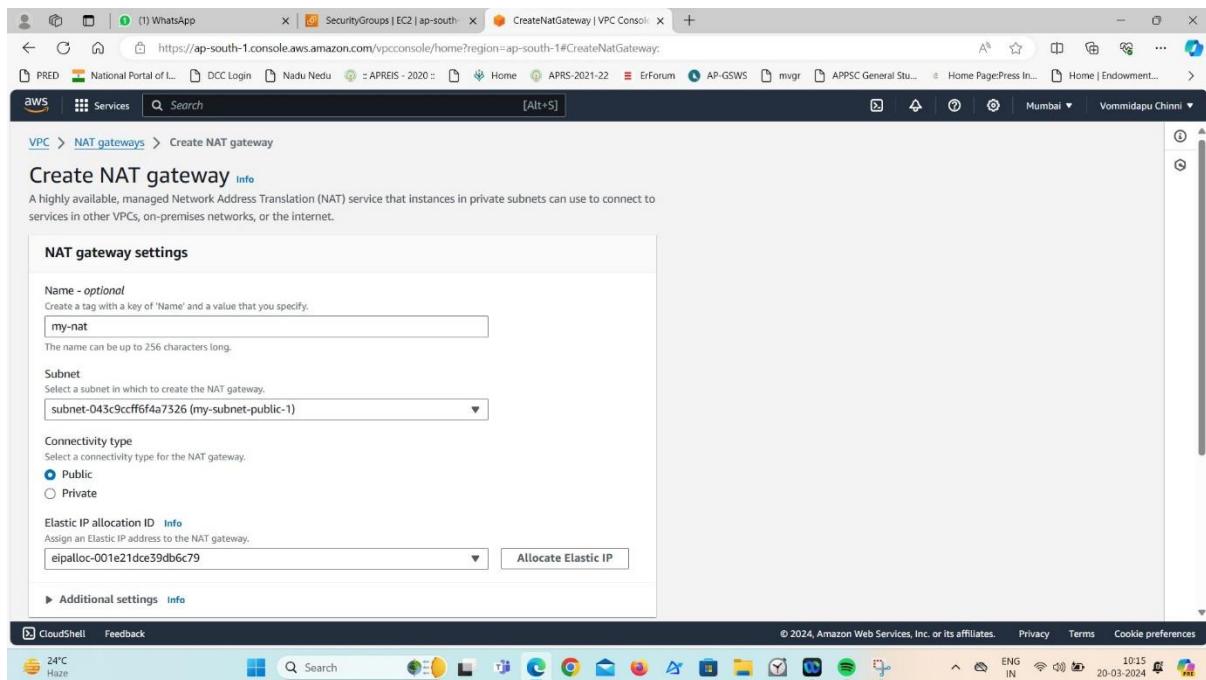
Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	Route table ID
my-subnet-private-3	subnet-0fdb345c9cf34e70	150.0.24.0/24	-	rtb-098ecfc8550bbd22a / my-routeta...
my-subnet-private-1	subnet-0e1bf9607ad2d1b57	150.0.22.0/24	-	rtb-098ecfc8550bbd22a / my-routeta...
my-subnet-public-1	subnet-043c9ccff64a7326	150.0.20.0/24	-	rtb-05d671248cecef70 / my-routeta...
my-subnet-public-2	subnet-009125c4cf54052ee	150.0.21.0/24	-	rtb-05d671248cecef70 / my-routeta...
my-subnet-private-2	subnet-06118ba884b21366b	150.0.23.0/24	-	rtb-098ecfc8550bbd22a / my-routeta...
my-subnet-private-4	subnet-0b406c4c29dc4489f	150.0.25.0/24	-	rtb-098ecfc8550bbd22a / my-routeta...

Selected subnets

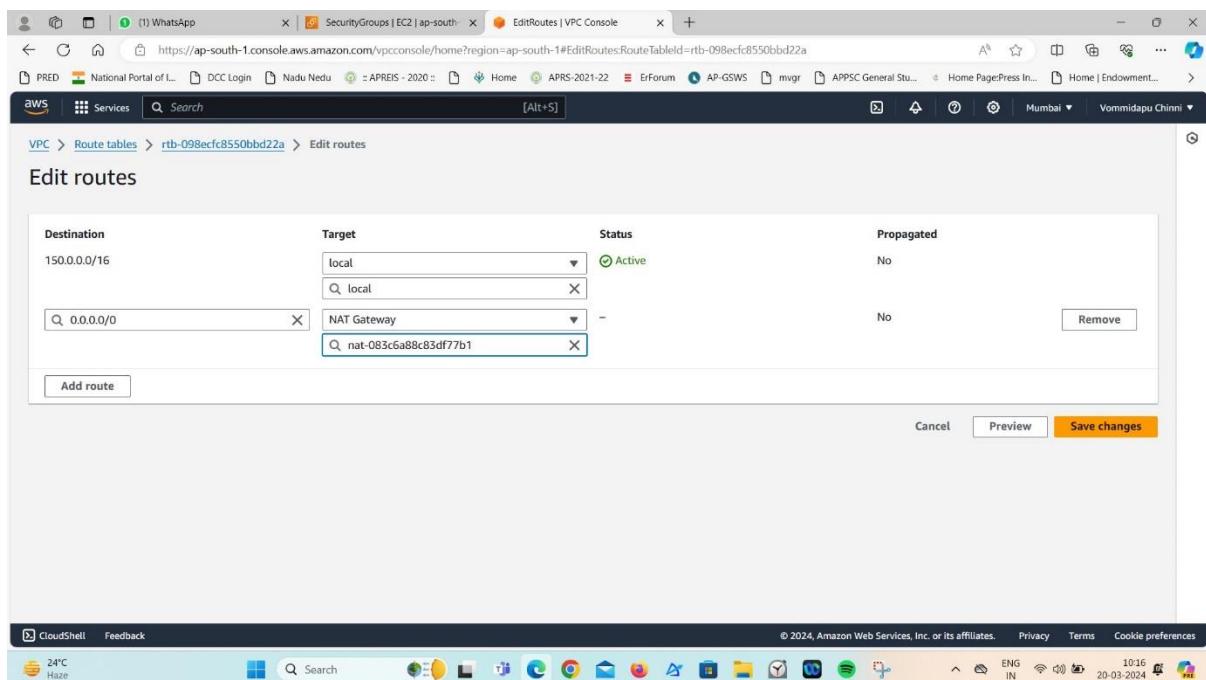
- subnet-0b406c4c29dc4489f / my-subnet-private-4
- subnet-0fdb345c9cf34e70 / my-subnet-private-3
- subnet-0e1bf9607ad2d1b57 / my-subnet-private-1
- subnet-06118ba884b21366b / my-subnet-private-2

Cancel **Save associations**

- Create NAT gateway, give name as my-nat.
- Select public subnet(my-subnet-public-1).
- Select connectivity type as IPV4.
- Click on allocate Elastic IP.



- Now go to private route and click on actions.
- Click on edit routes and add route.
- Give all traffic (0.0.0.0/0) and select NAT gateway.



Step2: Launch an EC2 instance.

- Go to EC2 dashboard click on launch instance.
- Name as project-ec2 and select ami as ubuntu.

- Instance type as t2.micro and key pair as project.
- Click on edit network settings, select our VPC and public subnet.
- Auto assign IP enable and create a security group as project-sg.
- Launch the instance.

Name and tags

Name: project-ec2

Application and OS Images (Amazon Machine Image)

Software Image (AMI): Amazon Linux 2023 AMI 2023.3.2...
ami-013168dc3850ef002

Virtual server type (instance type): t2.micro

Firewall (security group): New security group

Storage (volumes): 1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro) in the Regions in which you launch instances.

Launch instance

Application and OS Images (Amazon Machine Image)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below.

Search our full catalog including 1000s of application and OS images

Recents | My AMIs | **Quick Start**

Amazon Machine Image (AMI)

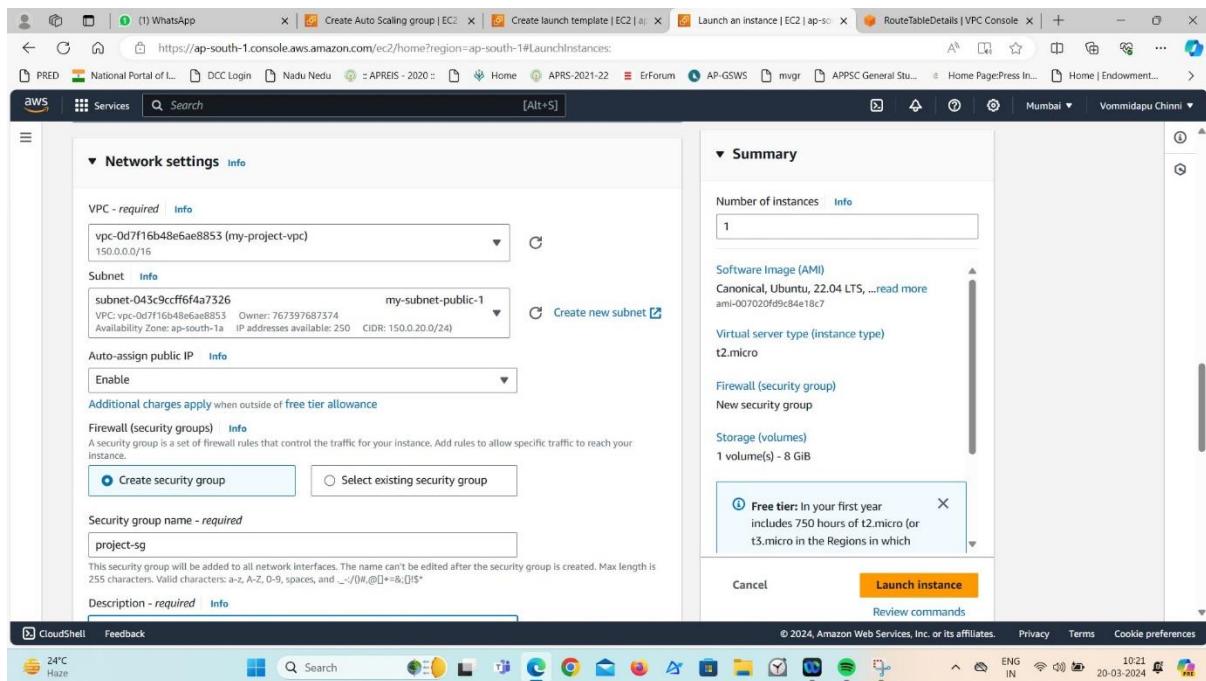
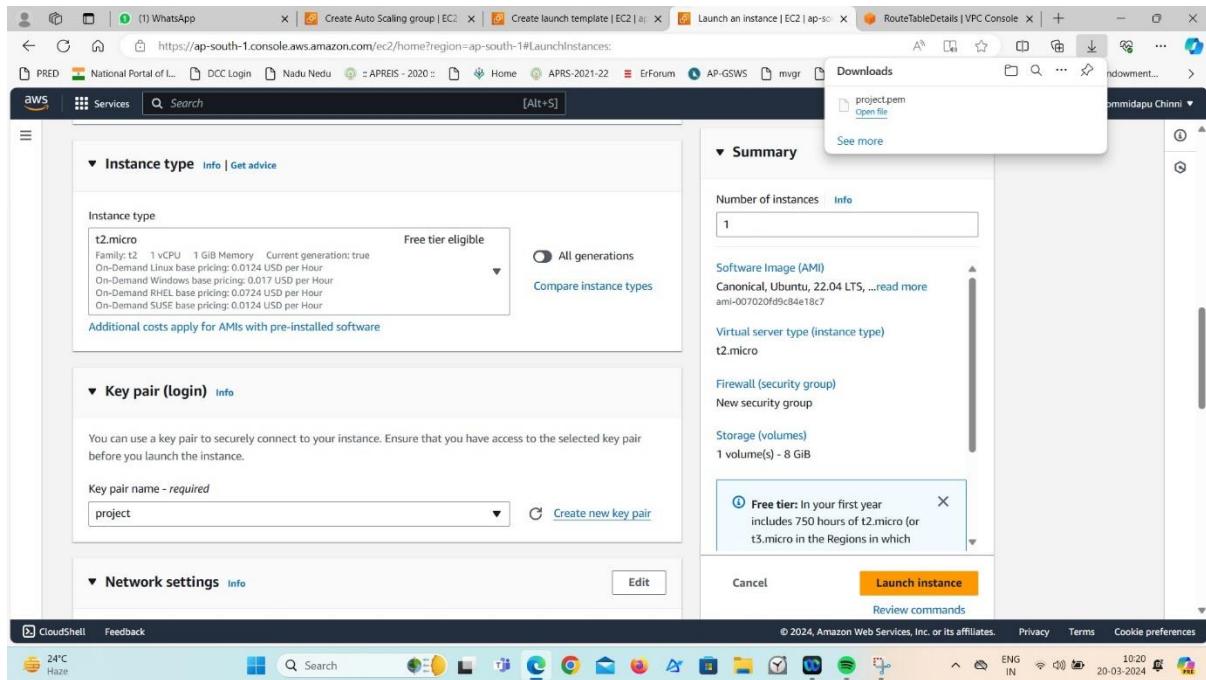
Ubuntu Server 22.04 LTS (HVM), SSD Volume Type
ami-0070209d9c3d4e18c7 (64-bit (x86)) / ami-09c443d9277298026 (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs

Description
Canonical, Ubuntu, 22.04 LTS, amd64 jammy image build on 2024-03-01

Architecture: AMI ID

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro) in the Regions in which you launch instances.

Launch instance



- Create another instance with private subnet. Through using public instance, we connect to the private instance.
- Then connect that instance we cannot connect after giving the project.pem, chmod and ssh -i and public ip of private instance.

```
https://ap-south-1.console.aws.amazon.com/ec2-instance/connect/ssh?connType=standard&instanceId=i-0dc522526c57cc086&cosUser=ubuntu&region=ap-south-1_&__vod=1

Welcome to Ubuntu 22.04.4 LTS (GNU/Linux 6.5.0-1014-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

System information as of Wed Mar 20 06:50:53 UTC 2024

System load:  0.0          Processes:           98
Usage of /:   34.3% of 7.57GB  Users logged in:  0
Memory usage: 61%           IPv4 address for eth0: 150.0.20.71
Swap usage:   0%           

Expanded security Maintenance for Applications is not enabled.

0 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

*** System restart required ***
Last login: Wed Mar 20 06:21:29 2024 from 13.233.177.4
ubuntu@ip-150-0-20-71:~$ sudo -i

i-0dc522526c57cc086 (project-ec2)
PublicIPs: 13.234.114.217 PrivateIPs: 150.0.20.71
```



```
[4] Wi | Auto | Launc | subic | Conn | EC2 In | Data | Auto | Subs | Insert | MySQ | MySQL | ...  
https://ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-0dc522526c57cc008&osUser=ubuntu&region=ap-south-1...  
  
aws Services Search [Alt+S]  
Last Login: Wed Mar 20 06:21:29 2024 from 13.233.177.4  
ubuntu@ip-150-0-20-71:~$ sudo -i  
root@ip-150-0-20-71:~# chmod 400 "project.pem"  
root@ip-150-0-20-71:~# ssh -i "project.pem" ubuntu@ec2-i-13-232-82-116.ap-south-1.compute.amazonaws.com  
Welcome to Ubuntu 22.04.4 LTS (GNU/Linux 6.5.0-1014-aws x86_64)  
  
* Documentation: https://help.ubuntu.com  
* Management: https://landscape.canonical.com  
* Support: https://ubuntu.com/pro  
  
System information as of Wed Mar 20 06:51:08 UTC 2024  
  
System load: 0.2080078125 Processes: 102  
Usage of /: 20.7% of 7.57GB Users logged in: 0  
Memory usage: 22% IPv4 address for eth0: 150.0.22.200  
Swap usage: 0%  
  
Expanded Security Maintenance for Applications is not enabled.  
0 updates can be applied immediately.  
  
Enable ESM Apps to receive additional future security updates.  
See https://ubuntu.com/esm or run: sudo pro status  
  
i-0dc522526c57cc008 (project-ec2)  
Public IP: 13.233.11.4.233 Private IP: 150.0.22.200
```



```
https://ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-0dc522526c57cc086&ostUser=ubuntu&region=ap-south-1... ...
PRED National Portal of L... DCC Login Nadu Nedu APRS-2020= Home APRS-2021-2022 ErForum AP-GSWs mvgr APPSC General Stu... Home Page|Press In... Mumbai Vommidiappu Chinni
[Alt+S]
Memory usage: 22% IPv4 address for eth0: 150.0.22.200
Swap usage: 0%
Expanded Security Maintenance for Applications is not enabled.
0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

Last login: Wed Mar 20 06:50:05 2024 from 150.0.20.71
root@ip-150-0-22-200:~# ping google.com
PING google.com (142.250.192.46) 56(84) bytes of data.
64 bytes from bom1s15-in-f14.1e100.net (142.250.192.46): icmp_seq=1 ttl=50 time=1.69 ms
64 bytes from bom1s15-in-f14.1e100.net (142.250.192.46): icmp_seq=2 ttl=50 time=1.48 ms
64 bytes from bom1s15-in-f14.1e100.net (142.250.192.46): icmp_seq=3 ttl=50 time=1.54 ms
64 bytes from bom1s15-in-f14.1e100.net (142.250.192.46): icmp_seq=4 ttl=50 time=1.53 ms
[1]+  Stopped                  ping google.com
root@ip-150-0-22-200:~# 
```



Step3: Create an AMI (image)

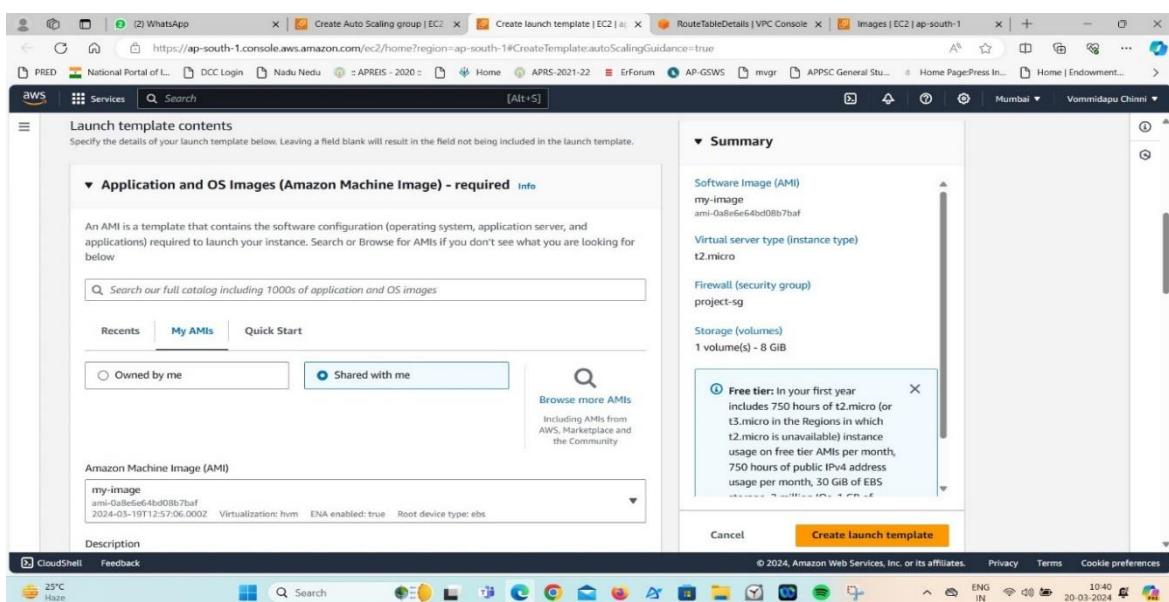
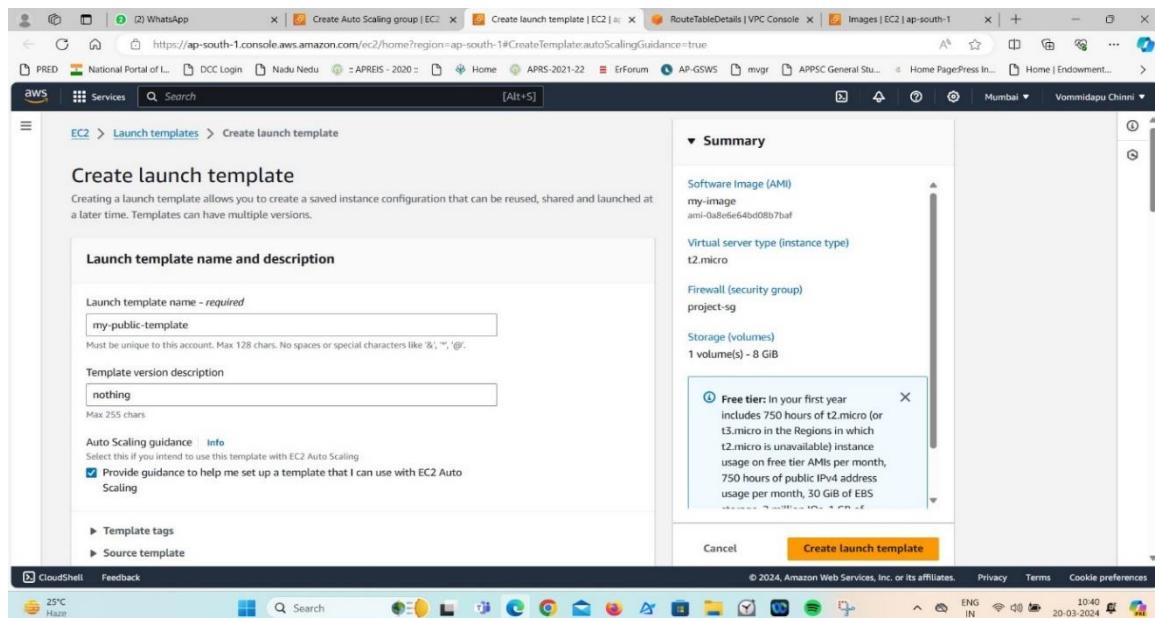
- After running the instance, click on actions.
- Click on image and templates and click on create image.
- Give image name as my-image.
- Wait until the image is available.

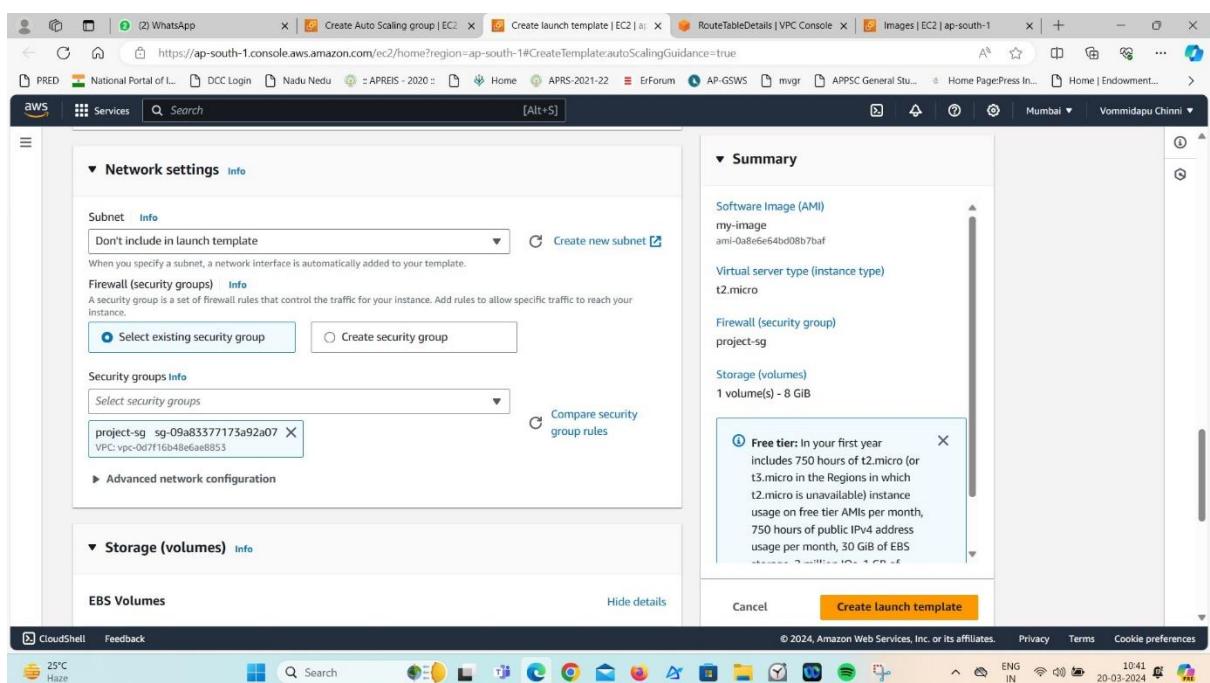
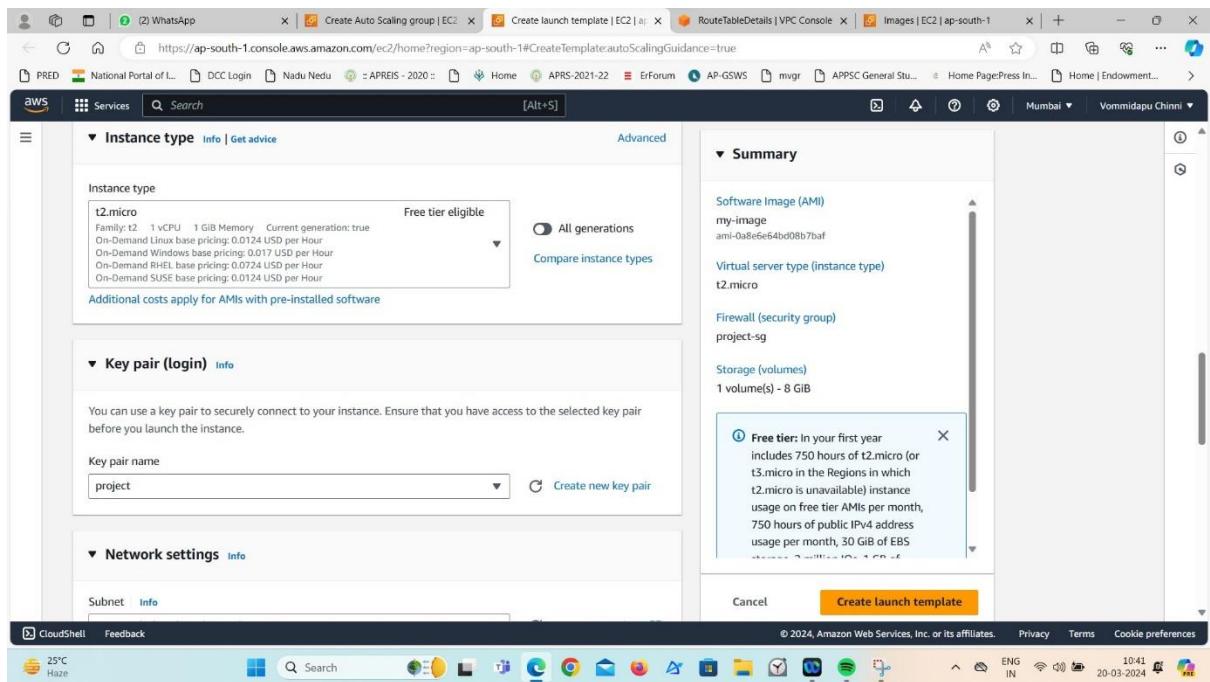
The screenshot shows the AWS EC2 Instances page. A single instance, 'project-ec2', is listed as 'Running'. In the Actions menu, the 'Image and templates' option is highlighted. Below the instance table, there are three buttons: 'Create image', 'Create template from instance', and 'Launch more like this'.

The screenshot shows the 'Create image' wizard. Step 1: Create image info. It displays the instance ID 'i-0dc522526c57cc086 (project-ec2)' and the image name 'my-image'. There is also a field for 'Image description - optional' which is currently empty. At the bottom, there are checkboxes for 'No reboot' (unchecked) and 'Enable' (unchecked).

Step4: Create Autoscaling group.

- For creating autoscaling group we need to create an launch template.
- After available of image. Click on create a launch template.
- Template name as my-public-template, description as nothing.
- Select AMI's as share with me, select my-image.
- Instance type as t2.micro and key pair as project.
- Select existing security group (project-sg) which is used to launch an EC2 instance.
- Now click on create launch template.





- Open autoscaling group.
- Click on create autoscaling group.
- Give name as public-asg.
- Choose the created launch template (my-public-template) and click on next.
- Select our VPC (my-project-vpc), and both public subnets.
- Click on next and click on no load balancer.

- Give desired capacity as 2 in sizing desired capacity min – 2 and max – 5 and click on next.
- Click on add notification, give name as my-topic and give email.
- Click on next and click on create auto scaling group.

Choose launch template

Specify a launch template that contains settings common to all EC2 instances that are launched by this Auto Scaling group.

Name

Auto Scaling group name
Enter a name to identify the group.
public-asg

Must be unique to this account in the current Region and no more than 255 characters.

Launch template

For accounts created after May 31, 2023, the EC2 console only supports creating Auto Scaling groups with launch templates. Creating Auto Scaling groups with launch configurations is not recommended but still available via the CLI and API until December 31, 2023.

Launch template
Choose a launch template that contains the instance-level settings, such as the Amazon Machine Image (AMI), instance type, key pair, and security groups.
my-public-template

Review

Launch template
Choose a launch template that contains the instance-level settings, such as the Amazon Machine Image (AMI), instance type, key pair, and security groups.
my-public-template

Create a launch template version [\[Create\]](#)

Version
Default (1)

Description
nothing

Launch template
my-public-template [\[Edit\]](#)
lt-0699bbc5b9ecf4d8

Instance type
t2.micro

AMI ID
ami-0a8e6e64bd08b7baf

Security groups
-

Request Spot Instances
No

Key pair name
project

Security group IDs
sg-09a83377175a92a07

Additional details

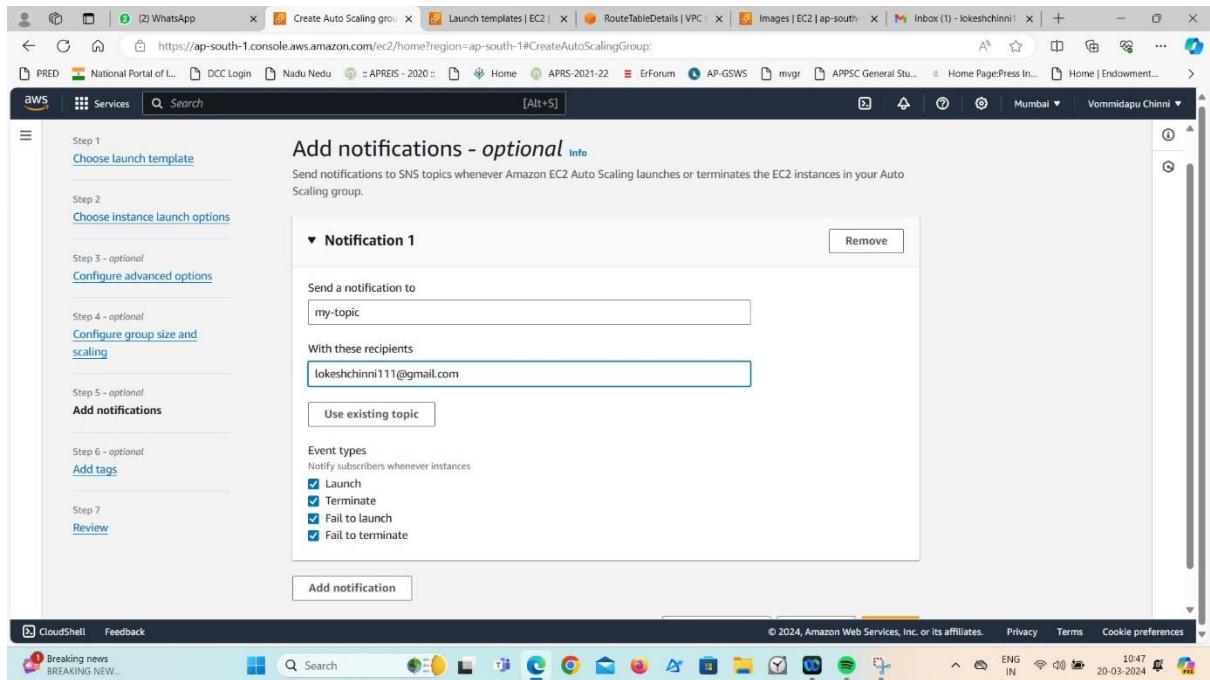
Storage (volumes)
-

Date created
Wed Mar 20 2024 10:41:47
GMT+0530 (India Standard Time)

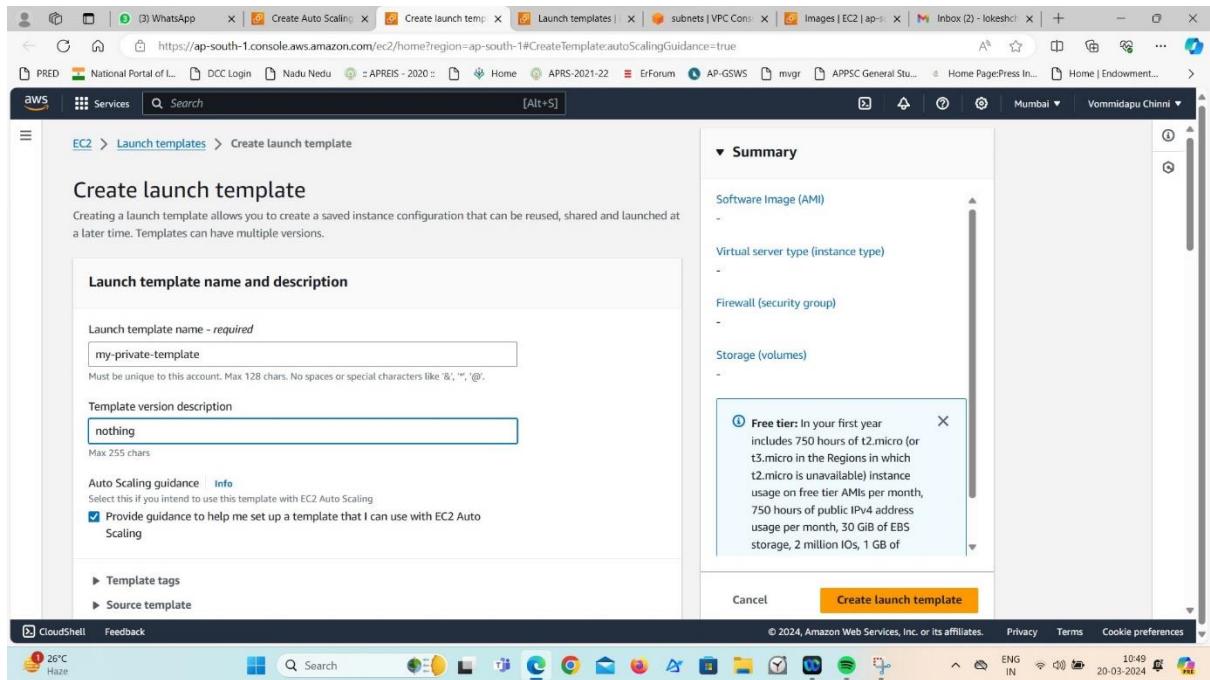
Screenshot of the AWS CloudShell interface showing the creation of an Auto Scaling group. The user is on Step 6 - optional 'Add tags'. The 'Network' section is active, showing the selected VPC (vpc-0d7f16b43e6ae6853) and two availability zones (ap-south-1a and ap-south-1b). The 'Next' button is highlighted.

Screenshot of the AWS CloudShell interface showing the creation of an Auto Scaling group. The user is on Step 1 'Choose launch template'. The 'Configure advanced options - optional' section is active, showing load balancing options (No load balancer, Attach to an existing load balancer, Attach to a new load balancer) and health checks (EC2 health checks, Always enabled). The 'Next' button is highlighted.

Screenshot of the AWS CloudShell interface showing the creation of an Auto Scaling group. The user is on Step 2 'Choose instance launch options'. The 'Configure group size and scaling' section is active, showing group size (Desired capacity: 2, Min desired capacity: 2, Max desired capacity: 5) and scaling limits (Min desired capacity: 2, Max desired capacity: 5). The 'Next' button is highlighted.



- In similar way create another launch template name as my-private-template and create auto scaling group name as private-asg.
- In it select create VPC and give two private subnets.



Screenshot of the AWS CloudFormation console showing the creation of a launch template. The 'Launch templates' tab is selected. A modal window titled 'Create launch template' is open, showing the configuration for a new launch template named 'my-launch-template'. The 'Software Image (AMI)' section shows 'my-image' (ami-0274dc33b334bd83f) selected. The 'Virtual server type (instance type)' is set to 't2.micro'. The 'Storage (volumes)' section indicates '1 volume(s) - 8 GiB'. A tooltip for the 'Free tier' is displayed, stating: 'In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS usage per month'. The 'Create launch template' button is highlighted.

Screenshot of the AWS CloudFormation console showing the creation of a launch template. The 'Launch templates' tab is selected. A modal window titled 'Create launch template' is open, showing the configuration for a new launch template named 'my-launch-template'. The 'Instance type' section shows 't2.micro' selected, with a note that it is 'Free tier eligible'. The 'Virtual server type (instance type)' is set to 't2.micro'. The 'Storage (volumes)' section indicates '1 volume(s) - 8 GiB'. A tooltip for the 'Free tier' is displayed, stating: 'In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS usage per month'. The 'Create launch template' button is highlighted.

Screenshot of the AWS CloudFormation console showing the creation of a launch template. The 'Launch templates' tab is selected. A modal window titled 'Create launch template' is open, showing the configuration for a new launch template named 'my-launch-template'. The 'Network settings' section shows 'Subnet' set to 'Don't include in launch template'. The 'Firewall (security groups)' section shows 'Select existing security group' selected, with 'project-sg' chosen. The 'Security groups info' dropdown shows 'project-sg sg-09a83377173a92a07'. The 'Storage (volumes)' section indicates '1 volume(s) - 8 GiB'. A tooltip for the 'Free tier' is displayed, stating: 'In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS usage per month'. The 'Create launch template' button is highlighted.

Screenshot of Step 2: Choose instance launch options. The 'Name' field contains 'private-asg'. A note states: 'For accounts created after May 31, 2023, the EC2 console only supports creating Auto Scaling groups with launch templates. Creating Auto Scaling groups with launch configurations is not recommended but still available via the CLI and API until December 31, 2023.' The 'Launch template' dropdown shows 'my-private-template'. The 'Version' dropdown shows 'Default (1)'. Buttons for 'Create a launch template' and 'Create a launch template version' are visible.

Screenshot of Step 3: Configure advanced options. The 'Launch template' dropdown shows 'my-private-template'. The 'Version' dropdown shows 'Default (1)'. The 'Description' field contains 'nothing'. The 'AMI ID' field shows 'ami-0274dc33b334bd83f'. The 'Key pair name' field shows 'project'. Under 'Additional details', the 'Date created' field shows 'Wed Mar 20 2024 10:51:47 GMT+0530 (India Standard Time)'. Buttons for 'Cancel' and 'Next' are at the bottom.

Screenshot of Step 6: Network. The 'VPC' dropdown shows 'vpc-0d716b48e6ae8853 (my-project-vpc)'. The 'Availability Zones and subnets' dropdown shows 'Select Availability Zones and subnets'. Two subnets are listed: 'ap-south-1a | subnet-0e1bf9607ad2d1b57 (my-subnet-private-1)' and '150.0.22.0/24'. Another subnet is listed: 'ap-south-1b | subnet-06118ba884b21366b (my-subnet-private-2)' and '150.0.23.0/24'. A 'Create a subnet' button is present. Buttons for 'Cancel', 'Skip to review', 'Previous', and 'Next' are at the bottom.

Screenshot of the AWS CloudShell interface showing the configuration of advanced options for an Auto Scaling group. The page displays three optional steps: Load balancing, Health checks, and Additional health check types.

Load balancing

Three options are available:

- No load balancer: Traffic to your Auto Scaling group will not be fronted by a load balancer.
- Attach to an existing load balancer: Choose from your existing load balancers.
- Attach to a new load balancer: Quickly create a basic load balancer to attach to your Auto Scaling group.

Health checks

Health checks increase availability by replacing unhealthy instances. When you use multiple health checks, all are evaluated, and if at least one fails, instance replacement occurs.

EC2 health checks: Always enabled

Additional health check types - optional

Screenshot of the AWS CloudShell interface showing the configuration of group size and scaling for an Auto Scaling group. The page displays seven optional steps: Group size, Desired capacity type, Scaling, Scaling limits, Automatic scaling, and Instance maintenance policy.

Group size

Set the initial size of the Auto Scaling group. After creating the group, you can change its size to meet demand, either manually or by using automatic scaling.

Desired capacity type

Choose the unit of measurement for the desired capacity value. vCPUs and Memory(GiB) are only supported for mixed instances groups configured with a set of instance attributes.

Scaling

You can resize your Auto Scaling group manually or automatically to meet changes in demand.

Scaling limits

Set limits on how much your desired capacity can be increased or decreased.

Screenshot of the AWS CloudShell interface showing the configuration of scaling limits and policies for an Auto Scaling group. The page displays five optional steps: Min desired capacity, Max desired capacity, Automatic scaling, and Instance maintenance policy.

Scaling limits

Set limits on how much your desired capacity can be increased or decreased.

Min desired capacity: 2

Max desired capacity: 5

Automatic scaling - optional

Choose whether to use a target tracking policy.

No scaling policies: Your Auto Scaling group will remain at its initial size and will not dynamically resize to meet demand.

Target tracking scaling policy: Choose a CloudWatch metric and target value and let the scaling policy adjust the desired capacity in proportion to the metric's value.

Instance maintenance policy - new

Control your Auto Scaling group's availability during instance replacement events. This includes health checks, instance refreshes, maximum instance lifetime features and events that happen automatically to keep your group balanced, called rebalancing events.

Control availability and cost during replacement events

An instance maintenance policy determines how much availability your application has when EC2 Auto Scaling replaces instances. It also establishes guardrails that limit the amount of capacity that

Screenshot of the AWS CloudShell interface showing the creation of an Auto Scaling group. The user is on Step 4 - optional, specifically adding notifications via SNS.

Add notifications - optional

Choose an SNS topic to use for notifications. The selected topic is "my-topic (lokeshchinni111@gmail.com)".

Event types

Notify subscribers whenever instances:

- Launch
- Terminate
- Fail to launch
- Fail to terminate

Review

Add notification

Buttons at the bottom: Cancel, Skip to review, Previous, Next.

Screenshot of the AWS CloudShell interface showing the creation of a Launch Template.

Launch Templates (1/2)

Launch Template ID	Launch Template Name	Default Version	Latest Version	Create Time	Created By
lt-00af2ee576bbdadf9	my-private-template	1	1	2024-03-20T05:21:47.000Z	arn:aws:iam::...
lt-0fde5d541d3a36216	my-public-template	1	1	2024-03-20T06:36:44.000Z	arn:aws:iam::...

Select a launch template

Buttons at the bottom: Actions, Create launch template.

Screenshot of the AWS CloudShell interface showing the creation of an Auto Scaling group.

Auto Scaling groups (2)

Name	Launch template/configuration	Instances	Status	Desired capacity	Min	Max	Available
private-asg	my-private-template Version Default	2	-	2	2	5	ap-sout...
public-asg	my-public-template Version Default	2	-	2	2	5	ap-sout...

0 Auto Scaling groups selected

Buttons at the bottom: Actions, Create Auto Scaling group.

- After creating autoscaling group we can get the four extra servers from both public and private auto scaling group.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP
i-056376eb29403f55	Running	t2.micro	2/2 checks passed	View alarms	ap-south-1b	-	
i-0d0725a521a9fd2b	Running	t2.micro	2/2 checks passed	View alarms	ap-south-1b	-	
project-ec2	Running	t2.micro	2/2 checks passed	View alarms	ap-south-1a	ec2-13-22	
i-05dbea3d95e182fd6	Running	t2.micro	2/2 checks passed	View alarms	ap-south-1a	-	
i-05ddc345456c9a4cd	Running	t2.micro	2/2 checks passed	View alarms	ap-south-1a	ec2-13-22	

Step5: Create subnet group

- Give name as my-subnet-grp and description nothing.
- Select created VPC.
- Give availability zones and select private subnets from each zone.
- Create the DB subnet group.

Create DB subnet group

To create a new subnet group, give it a name and a description, and choose an existing VPC. You will then be able to add subnets related to that VPC.

Subnet group details

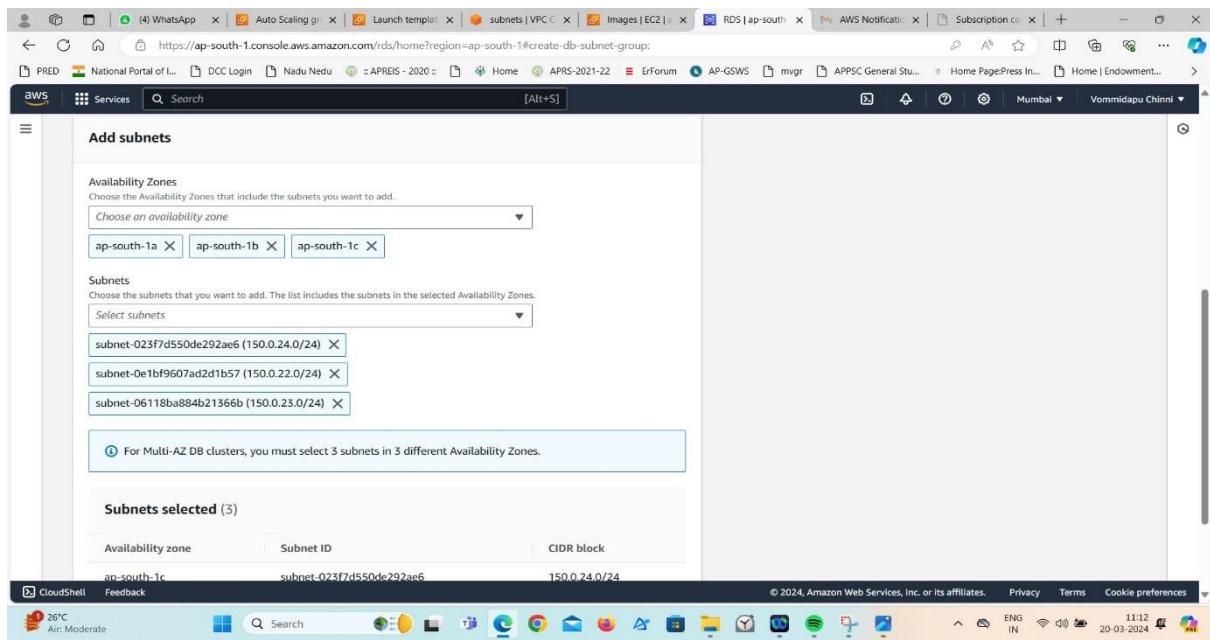
Name
You won't be able to modify the name after your subnet group has been created.

Must contain from 1 to 255 characters. Alphanumeric characters, spaces, hyphens, underscores, and periods are allowed.

Description

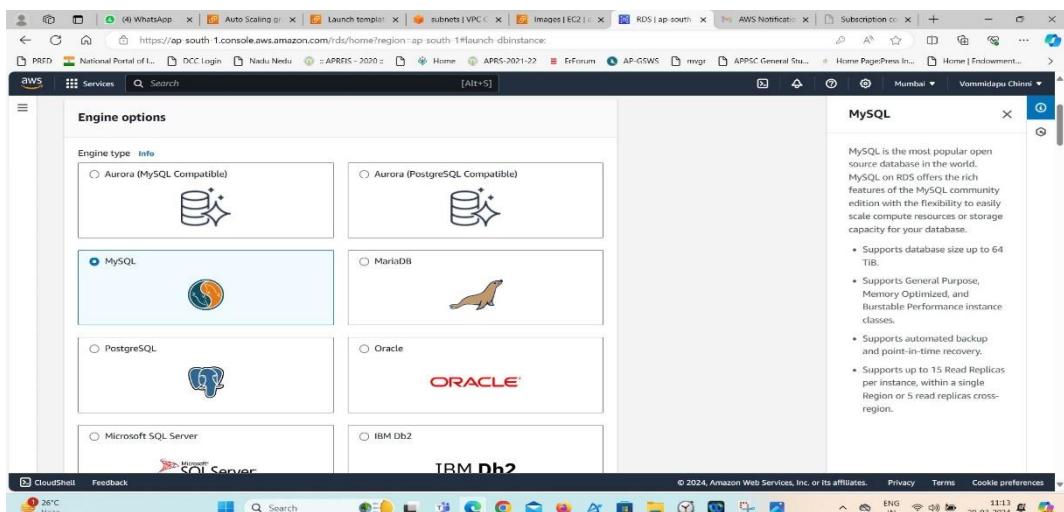
VPC
Choose a VPC identifier that corresponds to the subnets you want to use for your DB subnet group. You won't be able to choose a different VPC identifier after your subnet group has been created.

Add subnets



Step6: Create Database(RDS)

- Click on create database, select standard create, select engine type as MySQL.
- Select templates as production and select multi-AZ DB cluster.
- Select on self-managed, give password and confirm the password.
- Select memory optimized class.
- In connectivity, click on Don't connect to the EC2 compute resource and select created vpc (my-project-vpc).
- Select subnet group (my-subnet-grp) and give public access as yes.
- Choose existing security group (project-sg).
- Go to VPC dashboard, click on VPC, click on actions, go to edit VPC settings and click on the enable DNS hostnames.
- Create the database.



Templates

Choose a sample template to meet your use case.

- Production**
Use defaults for high availability and fast, consistent performance.
- Dev/Test**
This instance is intended for development use outside of a production environment.
- Free tier**
Use RDS Free Tier to develop new applications, test existing applications, or gain hands-on experience with Amazon RDS.
[Info](#)

Availability and durability

Deployment options [Info](#)
The deployment options below are limited to those supported by the engine you selected above.

- Multi-AZ DB Cluster**
Creates a DB cluster with a primary DB instance and two readable standby DB instances, with each DB instance in a different Availability Zone (AZ). Provides high availability, data redundancy and increases capacity to serve read workloads.
- Multi-AZ DB instance**
Creates a primary DB instance and a standby DB instance in a different AZ. Provides high availability and data redundancy; but the standby DB instance doesn't support connections for read workloads.
- Single DB instance**
Creates a single DB instance with no standby DB instances.

MySQL

MySQL is the most popular open source database in the world. MySQL on RDS offers the rich features of the MySQL community edition with the flexibility to easily scale compute resources or storage capacity for your database.

- Supports database size up to 64 TiB.
- Supports General Purpose, Memory Optimized, and Burstable Performance instance classes.
- Supports automated backup and point-in-time recovery.
- Supports up to 15 Read Replicas per instance, within a single Region or 5 read replicas cross-region.

Enter a name for your DB cluster. The name must be unique across all DB clusters owned by your AWS account in the current AWS Region.

database-1

The DB cluster identifier is case-insensitive, but is stored as all lowercase (as in "mydbcluster"). Constraints: 1 to 60 alphanumeric characters or hyphens. First character must be a letter. Can't contain two consecutive hyphens. Can't end with a hyphen.

Credentials Settings

Master username [Info](#)
Type a login ID for the master user of your DB cluster.

1 to 16 alphanumeric characters. The first character must be a letter.

Credentials management
You can use AWS Secrets Manager or manage your master user credentials.

- Managed in AWS Secrets Manager - most secure**
RDS generates a password for you and manages it throughout its lifecycle using AWS Secrets Manager.
- Self managed**
Create your own password or have RDS create a password that you manage.

Auto generate password
Amazon RDS can generate a password for you, or you can specify your own password.

Master password [Info](#)

Minimum constraints: At least 8 printable ASCII characters. Can't contain any of the following symbols: / \ * @

Confirm master password [Info](#)

MySQL

MySQL is the most popular open source database in the world. MySQL on RDS offers the rich features of the MySQL community edition with the flexibility to easily scale compute resources or storage capacity for your database.

- Supports database size up to 64 TiB.
- Supports General Purpose, Memory Optimized, and Burstable Performance instance classes.
- Supports automated backup and point-in-time recovery.
- Supports up to 15 Read Replicas per instance, within a single Region or 5 read replicas cross-region.

Instance configuration

The DB instance configuration options below are limited to those supported by the engine that you selected above.

DB instance class [Info](#)

Show instance classes that support Amazon RDS Optimized Writes [Info](#)
Amazon RDS Optimized Writes improves write throughput by up to 2x at no additional cost.

Standard classes (includes m classes)

Memory optimized classes (includes r classes)

db.r6gd.large (supports Amazon RDS Optimized Writes)
2 vCPUs 16 GiB RAM Network: 4,750 Mbps 118 GB NVMe SSD

Storage

Storage type [Info](#)
Provisioned IOPS SSD (io2) storage volumes are now available.

Provisioned IOPS SSD (io1)
Flexibility in provisioning I/O

Allocated storage [Info](#)
 GiB

MySQL

MySQL is the most popular open source database in the world. MySQL on RDS offers the rich features of the MySQL community edition with the flexibility to easily scale compute resources or storage capacity for your database.

- Supports database size up to 64 TiB.
- Supports General Purpose, Memory Optimized, and Burstable Performance instance classes.
- Supports automated backup and point-in-time recovery.
- Supports up to 15 Read Replicas per instance, within a single Region or 5 read replicas cross-region.

The screenshot shows the 'MySQL' configuration page in the AWS RDS console. On the left, under 'Connectivity & security', there are sections for 'Compute resource', 'Virtual private cloud (VPC)', 'DB subnet group', 'Public access', and 'Security groups'. In the 'Compute resource' section, two options are shown: 'Don't connect to an EC2 compute resource' (selected) and 'Connect to an EC2 compute resource'. The 'Virtual private cloud (VPC)' section shows 'my-project-vpc (vpc-0d7f16b48e6ae8853)' with '6 Subnets, 3 Availability Zones'. A note states: 'After a database is created, you can't change its VPC.' The 'DB subnet group' section shows 'my-subnet-grp' with '3 Subnets, 3 Availability Zones'. The 'Public access' section has 'Yes' selected. The right side of the screen displays a summary of MySQL features, including support for database sizes up to 64 TiB, various instance classes, automated backups, and up to 15 read replicas.

This screenshot shows the same MySQL configuration page as above, but with different settings. Under 'Public access', 'No' is selected. Under 'Security groups (firewall)', 'Choose existing' is selected, showing 'project-sg' as the chosen option. The right side of the screen remains the same, displaying MySQL features.

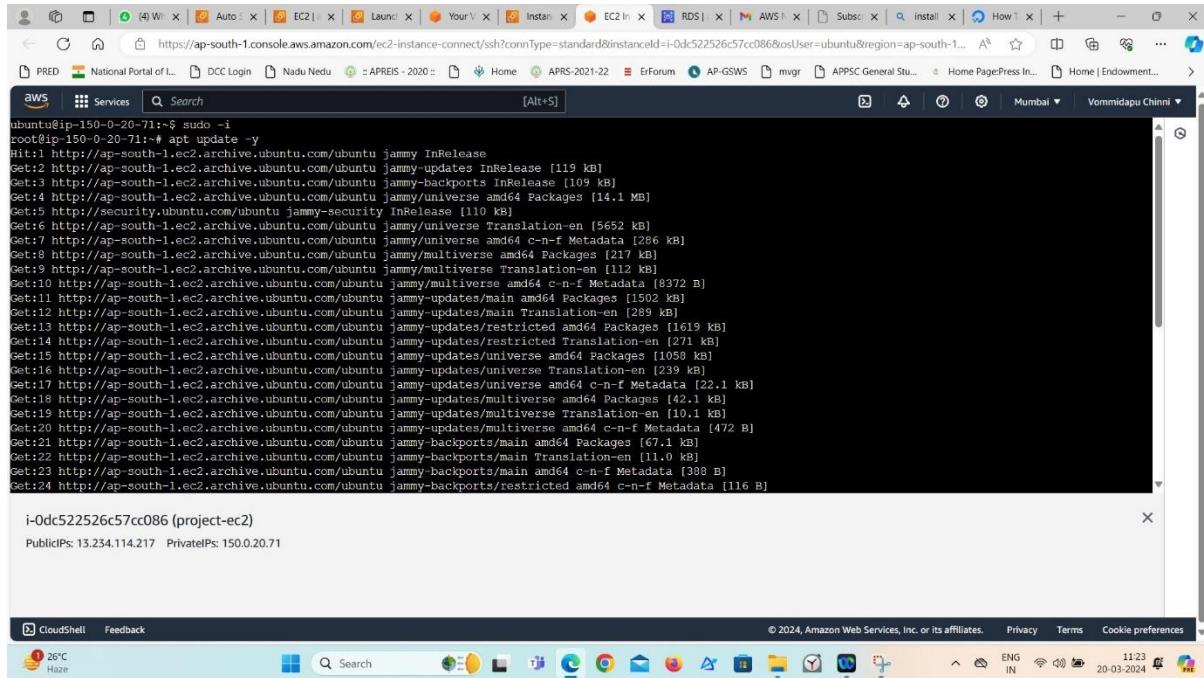
The screenshot shows the 'Edit VPC settings' page in the AWS VPC console. The 'VPC details' section shows the VPC ID as 'vpc-0d7f16b48e6ae8853' and the name as 'my-project-vpc'. The 'DHCP settings' section shows a single DHCP option set named 'dopt-0873a7fc1667054c'. The 'DNS settings' section has two checkboxes checked: 'Enable DNS resolution' and 'Enable DNS hostnames'. The 'Network Address Usage metrics settings' section is collapsed. The right side of the screen displays a summary of VPC features, including support for multiple subnets, automatic failover, and integration with AWS Lambda.

Step7: Establish connection

- Go to EC2 instance, click on project-ec2, open it click on connect and again click on connect.

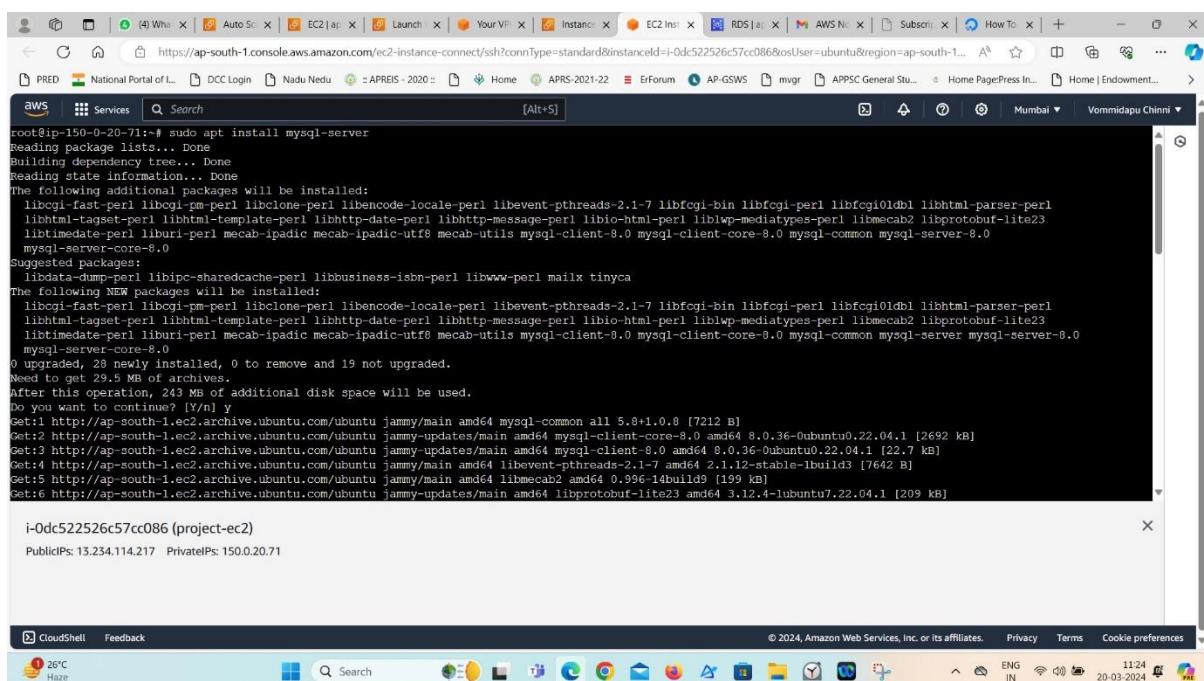
Give commands as:

1. sudo -i (convert from normal user to root user).
2. apt update -y
3. sudo apt install mysql-server (to install mysql)



```
ubuntu@ip-150-0-20-71:~$ sudo -i
root@ip-150-0-20-71:~# apt update -y
Hit:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease [109 kB]
Get:4 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 Packages [14.1 MB]
Get:5 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:6 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/universe Translation-en [5652 kB]
Get:7 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 c-n-f Metadata [286 kB]
Get:8 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/multiverse amd64 Packages [217 kB]
Get:9 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/multiverse Translation-en [112 kB]
Get:10 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/multiverse amd64 c-n-f Metadata [8372 B]
Get:11 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [1502 kB]
Get:12 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main Translation-en [289 kB]
Get:13 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packages [1619 kB]
Get:14 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted Translation-en [271 kB]
Get:15 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [1058 kB]
Get:16 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe Translation-en [239 kB]
Get:17 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 c-n-f Metadata [22.1 kB]
Get:18 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 Packages [42.1 kB]
Get:19 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse Translation-en [10.1 kB]
Get:20 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 c-n-f Metadata [472 B]
Get:21 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/main amd64 Packages [67.1 kB]
Get:22 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/main Translation-en [11.0 kB]
Get:23 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/main amd64 c-n-f Metadata [300 B]
Get:24 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/restricted amd64 c-n-f Metadata [116 B]

i-0dc522526c57cc086 (project-ec2)
Public IPs: 13.234.114.217 Private IPs: 150.0.20.71
```



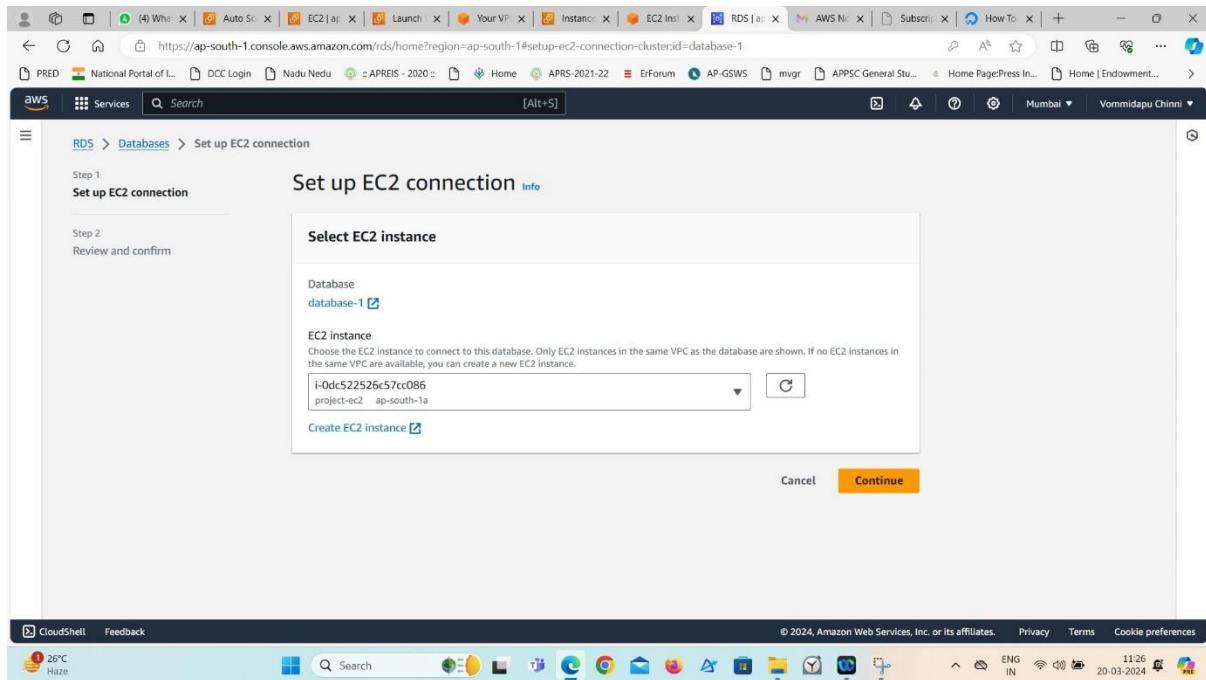
```
root@ip-150-0-20-71:~# sudo apt install mysql-server
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libcgi-fast-perl libcgi-pm-perl libclone-perl libencode-perl libevent-pthreads-2.1-7 libfcgi-bin libfcgi-perl libfcgioldbl libhtml-parser-perl
  libhtml-tagset-perl libhtml-template-perl libhttp-date-perl libhttp-message-perl libio-html-perl liblwp-mediatypes-perl libmecab2 libprotobuf-lite23
  libtimestr-perl liburi-perl mecab-ipadic mecab-ipadic-utf8 mecab-utils mysql-client-8.0 mysql-client-core-8.0 mysql-common mysql-server-8.0
  mysql-server-core-8.0
Suggested packages:
  libdata-dump-perl libipc-sharedcache-perl libbusiness-isbn-perl libwww-perl mailx tinyca
The following NEW packages will be installed:
  libcgi-fast-perl libcgi-pm-perl libclone-perl libencode-perl libevent-pthreads-2.1-7 libfcgi-bin libfcgi-perl libfcgioldbl libhtml-parser-perl
  libhtml-tagset-perl libhtml-template-perl libhttp-date-perl libhttp-message-perl libio-html-perl liblwp-mediatypes-perl libmecab2 libprotobuf-lite23
  libtimestr-perl liburi-perl mecab-ipadic mecab-ipadic-utf8 mecab-utils mysql-client-8.0 mysql-client-core-8.0 mysql-common mysql-server mysql-server-8.0
  mysql-server-core-8.0
0 upgraded, 28 newly installed, 0 to remove and 19 not upgraded.
Need to get 29.5 MB of archives.
After this operation, 243 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 mysql-common all 5.8+1.0.8 [7212 B]
Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 mysql-client-core-8.0 amd64 8.0.36-0ubuntu0.22.04.1 [2692 kB]
Get:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 mysql-client 8.0 amd64 8.0.36-0ubuntu0.22.04.1 [22.7 kB]
Get:4 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libevent-pthreads-2.1-7 amd64 2.1.12-stable-1build3 [7642 B]
Get:5 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libmecab2 amd64 0.996-14build9 [199 kB]
Get:6 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libprotobuf-lite23 amd64 3.12.4-lubuntu7.22.04.1 [209 kB]

i-0dc522526c57cc086 (project-ec2)
Public IPs: 13.234.114.217 Private IPs: 150.0.20.71
```

- Click on database-1, come down and click on set EC2 connection.
- Choose our instance (project-ec2).
- Click on continue and set up it.

The screenshot shows the AWS RDS Databases page. On the left, there's a sidebar with various navigation options like Dashboard, Databases, Query Editor, etc. The main area displays a table titled 'Databases (4)'. The table has columns for DB identifier, Status, Role, Engine, Region & AZ, Size, and Recommendation. One row is expanded to show three instances under 'database-1': database-1-instance-1 (Writer instance, MySQL Community, ap-south-1b, db.r6gd.large), database-1-instance-2 (Reader instance, MySQL Community, ap-south-1c, db.r6gd.large), and database-1-instance-3 (Reader instance, MySQL Community, ap-south-1a, db.r6gd.large). A modal window at the top right provides information about Blue/Green Deployments.

The screenshot shows the AWS RDS Compute Resources page. The sidebar includes options like Dashboard, Databases, and Compute resources. The main section displays a table for 'Connected compute resources (0)'. It shows two entries: database-1.cluster-cnkiu046ebfa.ap-south-1.rds.amazonaws.com (Writer, 3306) and database-1.cluster-ro-cnkiu046ebfa.ap-south-1.rds.amazonaws.com (Reader, 3306). Below this, there's a section for 'Proxies (0)' which is currently empty.



Now we can see that we are connected to MySQL server.

```
Last login: Wed Mar 20 06:09:38 2024 from 13.233.177.3
ubuntu@ip-150-0-20-71:~$ sudo -i
root@ip-150-0-20-71:~# sudo apt install mysql-server
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
mysql-server is already the newest version (8.0.36-0ubuntu0.22.04.1).
0 upgraded, 0 newly installed, 0 to remove and 19 not upgraded.
root@ip-150-0-20-71:~# sudo systemctl start mysql.service
root@ip-150-0-20-71:~# mysql -h database-1.cluster-cmkliu046ebfa.ap-south-1.rds.amazonaws.com -u admin -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 51
Server version: 8.0.35 Source distribution

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> [REDACTED]
i-0dc522526c57cc086 (project-ec2)
PublicIPs: 13.234.114.217 PrivateIPs: 150.0.20.71
```

Use commands like:

- show databases; (to show the list of databases are there)
- create database chinni; (to create a new database)
- use chinni; (server will take that database to use)
- create a table using this command
 - CREATE TABLE employee_table(
 - id int NOT NULL AUTO_INCREMENT,

- name varchar(45) NOT NULL,
 - occupation varchar(35) NOT NULL,
 - age int NOT NULL,
 - PRIMARY KEY (id)
 -);
 - we can insert data into that table using this command
 - `INSERT INTO table_name (column1, column2, column3, ...)`
`VALUES (value1, value2, value3, ...);`
 - show tables; (to show tables in that DB)
 - DROP is used to delete database and tables.
 - `DROP chinni;`

```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
+-----+
4 rows in set (0.00 sec)

mysql> create database chinni;
Query OK, 1 row affected (0.03 sec)

mysql> show databases;
+-----+
| Database |
+-----+
| chinni |
| information_schema |
| mysql |
| performance_schema |
| sys |
+-----+
5 rows in set (0.00 sec)

i-0dc522526c57cc086 (project-ec2)
PublicIPs: 13.234.114.217 PrivateIPs: 150.0.20.71
```

```
aws Services Search [Alt+S] PRED National Portal of... DCC Login Nadu Nedu :APREIS - 2020 Home APRS-2021-22 EForum AP-GWS APPSC General Stu... Home Page/Press In... Home | Endowment... Mumbai Vommidiappu Chinni

aws information_schema |
| mysql |
| performance_schema |
| sys |
+-----+
5 rows in set (0.00 sec)

mysql> use chinmi
Database changed
mysql> show tables
-> ^C
mysql> show tables
-> ^C
mysql> show tables;
Empty set (0.00 sec)

mysql> CREATE TABLE employee table(
-> id int NOT NULL AUTO_INCREMENT,
-> name varchar(45) NOT NULL,
-> occupation varchar(35) NOT NULL,
-> age int NOT NULL,
-> PRIMARY KEY (id)
-> );
Query OK, 0 rows affected (0.02 sec)

mysql> i-0dc522526c57cc086 (project-ec2)
PublicIP: 13.234.114.217 PrivateIP: 150.0.20.71

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences
```

```
aws Services Search [Alt+S]
https://ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-0dc522526c57cc086&osUser=ubuntu&region=ap-south-1... A
National Portal of... DCC Login Nadu Nedu APR-EIS - 2020 : Home APRS-2021-22 ErForum AP-GWS mgv APPSC General Stu... Home Page/Press In... Home | Endowment... Mumbai Vommidiapu Chinni
aws Services Search [Alt+S]
mysql> use chinni;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A
Database changed
mysql> show tables;
+-----+
| Tables_in_chinni |
+-----+
| employee_table |
+-----+
1 row in set (0.00 sec)

mysql> INSERT INTO employee_table (id, name, occupation, age) VALUES ('120', 'chinni', 'student', '20');
Query OK, 1 row affected (0.00 sec)

mysql> select *from employee_table;
+----+----+----+----+
| id | name | occupation | age |
+----+----+----+----+
| 120 | chinni | student | 20 |
+----+----+----+----+
1 row in set (0.00 sec)

mysql> i-0dc522526c57cc086 (project-ec2)
PublicIPs: 13.234.114.217 PrivateIPs: 150.0.20.71
```



```
aws Services Search [Alt+S]
https://ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-0dc522526c57cc086&osUser=ubuntu&region=ap-south-1... A
National Portal of... DCC Login Nadu Nedu APR-EIS - 2020 : Home APRS-2021-22 ErForum AP-GWS mgv APPSC General Stu... Home Page/Press In... Home | Endowment... Mumbai Vommidiapu Chinni
aws Services Search [Alt+S]
1 row in set (0.00 sec)

mysql> INSERT INTO employee_table (id, name, occupation, age) VALUES ('120', 'chinni', 'student', '20');
Query OK, 1 row affected (0.00 sec)

mysql> select *from employee_table;
+----+----+----+----+
| id | name | occupation | age |
+----+----+----+----+
| 120 | chinni | student | 20 |
+----+----+----+----+
1 row in set (0.00 sec)

mysql> ALTER TABLE employee_table ADD email_ID varchar(40) NOT NULL;
Query OK, 0 rows affected (0.02 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> select *from employee_table;
+----+----+----+----+
| id | name | occupation | age | email_ID |
+----+----+----+----+
| 120 | chinni | student | 20 |
+----+----+----+----+
1 row in set (0.00 sec)

mysql> i-0dc522526c57cc086 (project-ec2)
PublicIPs: 13.234.114.217 PrivateIPs: 150.0.20.71
```



```
aws Services Search [Alt+S]
https://ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-0dc522526c57cc086&osUser=ubuntu&region=ap-south-1... A
National Portal of... DCC Login Nadu Nedu APR-EIS - 2020 : Home APRS-2021-22 ErForum AP-GWS mgv APPSC General Stu... Home Page/Press In... Home | Endowment... Mumbai Vommidiapu Chinni
aws Services Search [Alt+S]
mysql> DROP Table employee_table;
Query OK, 0 rows affected (0.02 sec)

mysql> show tables;
Empty set (0.00 sec)

mysql> DROP database chinni;
Query OK, 0 rows affected (0.01 sec)

mysql> show database;
mysql> show database;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'database'
at line 1
mysql> show databases;
+-----+
| database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
+-----+
4 rows in set (0.00 sec)

mysql> i-0dc522526c57cc086 (project-ec2)
PublicIPs: 13.234.114.217 PrivateIPs: 150.0.20.71
```

